DARPA Urban Challenge

Technical Paper

April 13, 2007

Autonomous Solutions, Inc.

Sarah Gray	Sarah@AutonomousSolutions.com
Bruce Thorne	Bruce.Thorne@AutonomousSolutions.com
Zeke Susman	Zeke@AutonomousSolutions.com
Josh Johnston	Josh.Johnston@AutonomousSolutions.com
Tom Petroff	Tom@AutonomousSolutions.com
Dale Lord	Dale.Lord@AutonomousSolutions.com

Sarnoff Corporation

Jayan Eledath	jeledath@sarnoff.com
Mayank Bansal	mbansal@sarnoff.com
Gregory Kreutzer	gkreutzer@sarnoff.com
Aveek Das	adas@sarnoff.com
Oleg Naroditsky	onaroditsky@sarnoff.com

DeVivo AST, Inc.

Woody English	woodyenglish@DeVivoAST.com

1 Abstract

Team Autonomous Solutions is a Track 'A' competitor in the 2007 Defense Advanced Research Projects Agency (DARPA) Urban Challenge. Capitalizing on the experience of the team members in vehicle automation and sensing, the team established architecture based on existing hardware and software components thus limiting the impact of the requirements for urban travel to the Vehicle Intelligence module. Minor modifications to the position estimation component, the object detection processing, planning algorithms and low level vehicle control have been necessary to reach the current functional state, but the primary challenge remains in the refinement and testing of the automated driver.

Testing at both the system level and during integration and characterization of the architectural elements has resulted in further refinement of the overall design and simplification in the analysis of the problem space. One such refinement is the reduction in complexity of the perception system interface—only lane markings, vehicles and generic objects are reported. Another significant change from the initial design, one that is constrained to the Vehicle Intelligence module and results in further reduction in complexity, is the replacement of the decision tree control logic with the much simpler voter-arbiter approach in which reasoning objects make recommendations on a small set of primitive behaviors.

2 Introduction

Team Autonomous Solutions is comprised of Autonomous Solutions, Incorporated (ASI), Sarnoff Corporation, and DeVivo AST, Inc. As the prime contractor, Autonomous Solutions, Inc. brings years of experience in automating vehicles to the table. The software and hardware systems that ASI has developed have been refined through thousands of hours of use by government and commercial customers. ASI roles in the DARPA Urban Challenge are low-level vehicle automation, mission and path planning, developing an expert learning system, and overall program management. Sarnoff Corporation brings over 25 years of experience in the development and deployment of real-time vision systems for several applications, including robotic navigation and obstacle avoidance and automotive safety systems. Since 2000, Sarnoff has worked with several automotive Tier 1 suppliers and Original Equipment Manufacturers (OEMs) to develop real-time advanced stereo and mono-vision based systems. These systems identify the road surface and shape, and lane markers. They also detect and track vehicles, pedestrians and other obstacles on the road, as well as estimate the range, closing velocity and position of those obstacles. These systems have been tested extensively on automotive OEM vehicles. Sarnoff's role in the DARPA Urban Challenge is to provide the sensing capabilities required to navigate safely in traffic and to localize the vehicle in a world co-ordinate system. DeVivo Automated Systems Technology, Inc. experience ranges from integration and testing of unmanned ground vehicle systems, program coordination and management, and test and evaluation. DeVivo AST performs the Team Autonomous Solutions test direction and coordination functions.

For the DARPA Urban Challenge we have established a novel approach for autonomous unmanned systems operating in an urban environment that exploits existing technologies developed by the team including:

a) Sarnoff Corporation's vision and sensor fusion technology for determining the vehicle's position in GPS denied areas and for detection and classification of the static and dynamic objects in the environment

b) ASI's cost based high and low level mission planning software and their vehicle automation package.

These existing technologies, which are simple, reliable, and utilize off-the-shelf parts and technologies where possible, are currently being expanded and coupled with newly developed technologies including scalable driver emulation logic designed to obey the rules of the road safely in the presence of many other moving manned and unmanned vehicles.

3 Analysis

Analysis of the problem space as specified in the DARPA Urban Challenge Rules (December 11, 2006) result in the identification of eight primary challenges to the integration of an autonomous control system for urban travel. These challenges comprise the high level system requirements and map directly to the system architecture. The decomposition of the Urban Challenge Rules into this set of capabilities is based on both engineering judgment and the current set of software and hardware tools the team has developed prior to the Urban Challenge. Each identified challenge area is listed and discussed briefly below:

Sensing

Data capture from stereo vision systems and LIDAR (Light Imaging Detection and Ranging) is collected with the associated information including time and sensor position and orientation in a vehicle relative coordinate system. The primary sensing system on board the unmanned system is based on stereo imagery. At present, one forward looking stereo camera pair detects objects in the environment including vehicles, lane markings and other entities. All entities not categorized as vehicles or lane markings are considered obstacles. A list of sensed entities is sent to the vehicle intelligence processing module at the image capture frame rate. Current measured latencies have been approximated to 300 milliseconds, however optimization including the removal of built in test and debug statements has not been performed. The LIDAR provides a means to eliminate much of the latency with the vision system output by sending output directly to the Expert Driver (discussed below). The Expert Driver fuses the LIDAR data with the output of the vision system to determine correct behavior.

Object Detection

The data collected during the sensing process is analyzed. Objects are extracted from the data and reported to the vehicle intelligence processing and high level planning components as described above. Object detection from the stereo imagery provides both range and feature data to the planning processes. Lane markings are identified by their contrast and association with the ground plane. Vehicle characterization is more complex, but various common properties allow for the differentiation of them from other objects. Any object not characterized as either a lane marking or vehicle is reported as a generic obstacle.

Position Estimation

The determination of the location of the vehicle system in the environment is performed by fusing data from inertial measurement devices, Global Positioning System inputs, vehicle speed and turn rate, and Visual Odometry data. Visual Odometry provides input to the position determination by estimating positional differences in detected features between captured stereo images [2].

High Level Planning

The mission, as described by the Mission Data File (MDF) and Route Network Definition File (RNDF), is planned by the Autonomous Solutions Mobius™ product. During execution of the mission, Mobius™ map data is updated with the object detection data and vehicle status. Mobius™ will replan when the reactive planner, described below, cannot resolve the current command (reconnect to the original route) or when the vehicle is paused and moved off the route by a human driver or through tele-operation.

Reactive Planning

Upon the detection of an obstacle in the planned route, the Reactive Planner will establish an alternative route and speed around the object in an effort to remain as close as possible to the mission route. The reactive planner is one of the two software elements that comprise the Vehicle Intelligence Module.

Expert Driver

The Expert Driver is the other software element of the Vehicle Intelligence Module. The basic driving rules, as established through analysis of the DARPA Urban Challenge Rules, comprise the core logic of the Expert Driver. Interpretation of the objects as detected by the perception systems are weighed against the rules and control execution of vehicle mobility by the reactive planner. This capability does not perform any planning.

Vehicle Control

Low level vehicle control is provided by the Autonomous Solutions Neuron product. Neuron takes primitive driving commands such as the Joint Architecture for Unmanned Systems (JAUS) Wrench message and translates them to actuation commands. The Neuron processor also provides low level vehicle status back to the high level logic.

Vehicle Automation

The 2006 Hybrid Toyota Highlander SUV was selected for the Urban Challenge allowing Team Autonomous Solutions to utilize many of the OEM drive by wire capabilities via custom interface boards and OEM sensors via CAN bus protocol and the OBD II port to facilitate ease of automation. Interface boards are used to command OEM drive by wire systems to control the throttle and brake while a servo motor system is used for drive by wire steering. The steering system is placed in line with OEM steering to allow for OEM driving in manual mode. The transmission control uses a CAN driver to run a standard actuator which is inline with a transmission box enabling manual bypass of this actuator. A fail-safe stored energy system is used for redundancy in the emergency braking system.

The vehicle system provides the power and networking as required by the distributed computing system. An off the shelf power inverter is used to draw power from the vehicle's main 288 VDC battery which is charged by a generator connected directly to the vehicle's gasoline engine and inverted to a usable 5000 VA 120 VAC power distribution system. Both a high speed Ethernet network and a CAN bus network are used for communications between the on-board computing systems.

4 System Architecture

Team Autonomous Solutions' modular approach to the design of the autonomous control system has proven useful in isolation of capabilities, integration of previously developed components, and in integration testing. A JAUS based message set has been established to transfer data between the modules. Simulation versions of each module have been developed or in the case of Neuron and the Reactive Planner were already contained in Mobius™. The five modules include Mobius™, Neuron™, Vehicle Intelligence, Perception and the Sarnoff Video Inertial Navigation System (VINS). The relationships between the modules are shown in the High Level System Block Diagram below and further explained in the Design section.

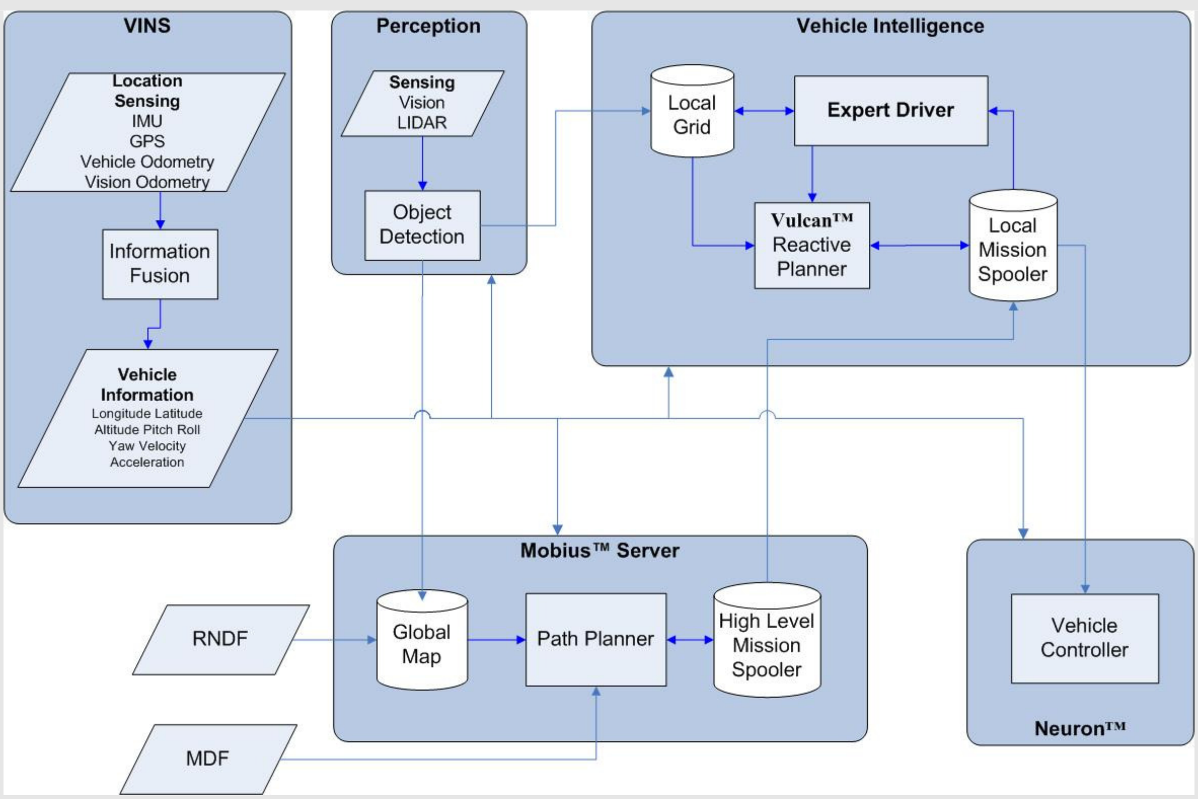

Figure 1, High Level System Block Diagram

The systems engineering approach taken by the team results in a direct map of the challenges identified into the system architecture. Figure 1, High Level System Block Diagram, represents the allocation of the challenges to the major functional modules. This diagram also shows the general flow of information used within each of the computational elements. The information flow depicted in the diagram is representative of data necessary for the decision processing, but does not address control flow.

The modularity of the design as presented in the block diagram above isolates the majority of the actual control logic for mobility to the Vehicle Intelligence module. The details of this control are discussed in the design section for the reactive planner and the expert driver below. This module receives high level plans from Mobius™ and position information and detected environmental elements from the Position Estimation and Perception modules respectively. The output from the Vehicle Intelligence module is a low level plan to the Neuron™ vehicle controller. The simplicity of this modular approach is shown in Figure 2, High Level System Control Diagram.

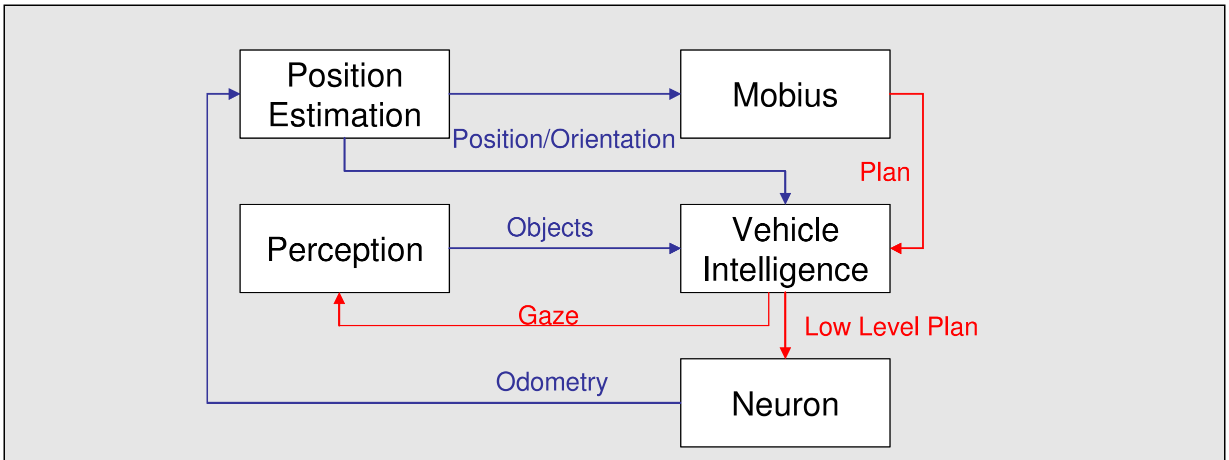

Figure 2, High Level System Control Diagram
The blue data flows represent information and the red arrows represent control.

The control flow and architecture are best described in terms of a portion of a mission execution. As shown in the graphic above, the Mobius™ module uses the currently reported position and after reading the Route Network Definition File and the Mission Data File, it generates the initial route plan. The plan is sent in large segments, to the mission spooler in the Vehicle Intelligence module. Vehicle Intelligence module inputs, aside from the mission plan, include the object reports from the Perception module and updates on vehicle position and orientation. The Vehicle Intelligence module commands the sensor gaze to optimize sensing output based on the current vehicle goals. The Vehicle Intelligence locally optimizes the mission plan based on the sensed information provided by the Perception module and then sends smaller sections of the local plan to the Neuron™ processor. Neuron™ converts the planned segments into low level mobility commands to accomplish the high level plan. This process is performed repeatedly with the exception of the generation of the high level plan which is only performed when the Vehicle Intelligence module cannot resolve a workable route to connect back to the planned path.

5 Design

The nature of the Urban Challenge, development of a highly complex system on a strained budget and brief schedule, forces design decisions to be made in less than ideal circumstances. However, the division in the development of the sensing systems and the intelligence software has forced the team to focus on efficient communications—the interfaces between the major components. This focus allowed the early identification of necessary data and guided the majority of design decisions. This section provides additional detail on the major design elements within the Team Autonomous Solutions Urban Challenge entry and discusses characterization results of these elements where appropriate.

5.1 Perception

The Sarnoff-developed stereovision-based perception module is designed to provide full 360° coverage of the area surrounding the autonomous vehicle. The hardware components used to implement the perception module are all Commercial-Off-The-Shelf (COTS) products. For vision processing, we use five (5) Intel Core 2 Duo Quad-core processors with 4 GB RAM, 6 Matrox Odyssey XPRO+ PCI-X capture cards and 6 Imperx IPX-2M30H-L stereo pairs (12 cameras total), each with a resolution of 1920x1080 and 33 frames per second (fps) CameraLink output.

5.1.1 Sensor placement

The 6 stereo pairs are mounted on the vehicle roof-rack to provide full 360° coverage as depicted in Figure 3, Sensor Placement.

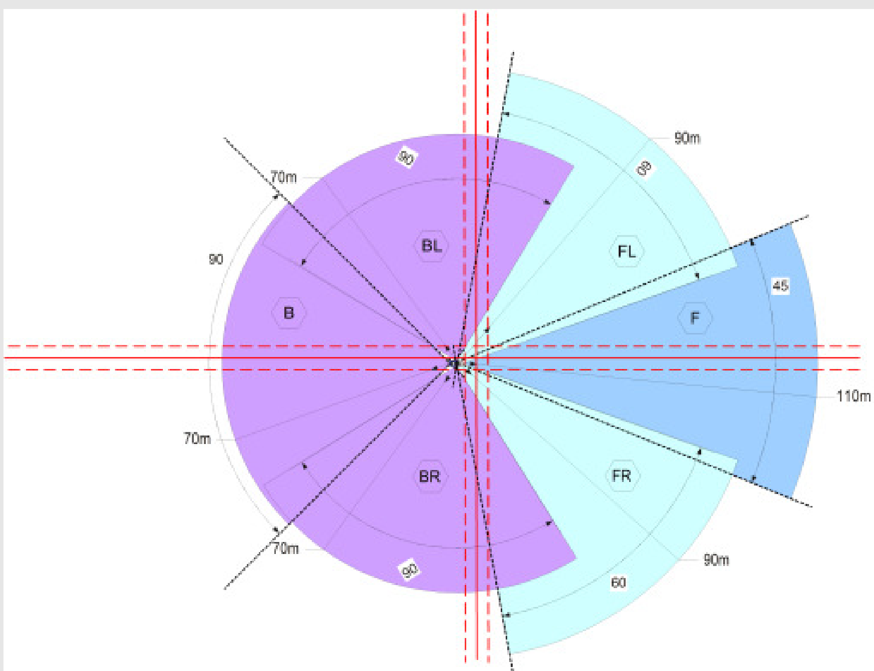

Figure 3, Sensor Placement
By using a combination of three different field-of-view sensors, we are able to cover different distances in different directions. Blue = 45º (110m), Cyan = 60º (90 m), Purple = 90º (70 m).

5.1.2 Stereo-based Obstacle Detection and Tracking

The stereo-based obstacle detection and tracking algorithms have two components:

1) Short-range geometric obstacle detection and mapping, and
2) Long-range vehicle detection and tracking.

5.1.2.1 Short-Range Vehicle and Obstacle Detection and Mapping

A block diagram of the obstacle detector is shown in Figure 4, Block Diagram of Stereo-Based Obstacle Detector and Map Generator, below.

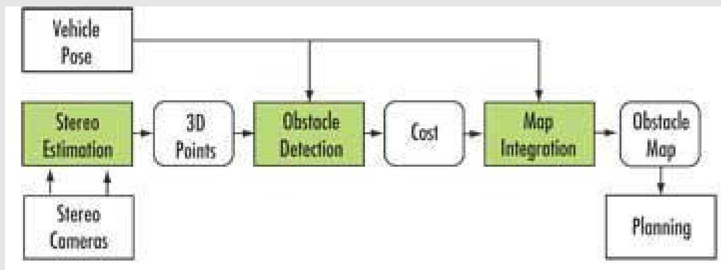

Figure 4, Block Diagram of Stereo-Based Obstacle Detector and Map Generator

To get the widest field-of-view for obstacle detection while maintaining a good throughput, we sub sample the input 1920x1080 image to a size of 640x360 before running the stereo estimator. The stereo estimator module computes a dense disparity image using local correlation between the left and right (stereo) images for 32 horizontal shift positions, selects the minimum and interpolates the data to compute a sub-pixel disparity value for every pixel in the image. It also computes a texture energy measure for each pixel in the image that can be used to mask low texture data in the image. A left-right checking module is used to mask the occlusion boundaries and to increase the reliability of the disparity data. In our implementation for the DARPA Urban Challenge, the stereo estimator is implemented on the Matrox Odyssey XPRO+. To provide high-speed processing capabilities, the Matrox Odyssey board incorporates a powerful G4 PowerPC microprocessor, a 64 element pixel accelerator (PA), 2GB on-board memory, and a customizable Processing FPGA. The stereo estimator is implemented using the high level Odyssey Native Library (API calls), which provides direct access to the arithmetic and logic functionality of the PA. To best use the board's capabilities, we pipeline the stereo computation so that the FPGA is busy rectifying the input images at time *t* while the PA is computing the dense disparity image for the input at time *t-1*.

The obstacle detection (OD) function analyzes stereo derived range data together with vehicle attitude to associate a traversal cost with the 3D structure discovered by the stereo process. The traversal cost is in the form of a height (above a reference "ground" surface) cost and a slope (local change relative to reference surface) cost. The algorithm compensates for ground resolution variation, pre-filters the range image to reduce noise and analyzes the data in X-Z co-ordinates (creates a map). This analysis uses a multi-resolution technique to estimate a function Y_{ref} that is used to compute the height cost map and a slope map for each X-Z location. The spread of heights at a single X-Z location and differences in lowest elevation between neighboring pixels classifies each map point as an obstacle or not. Map pixels are aggregated to obtain cells of a constant size on the ground near the camera and used to estimate the slope.

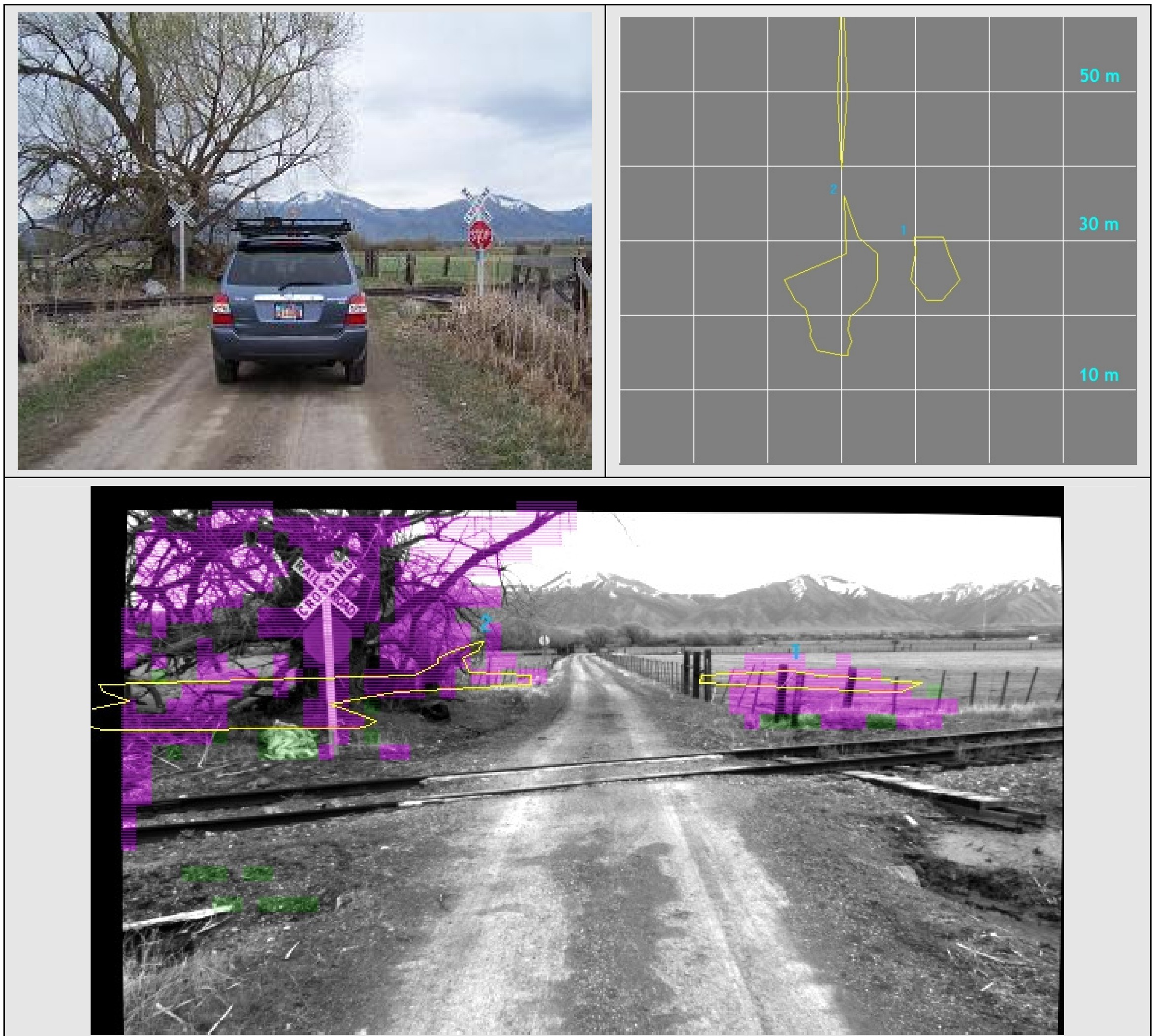

Figure 5, Obstacles Detected and Grouped by OD

Figure 5, Obstacles Detected and Grouped by OD, shows a sample OD result from an experiment conducted on the Urban Challenge vehicle near the Autonomous Solutions site in Logan, Utah.

The Map Integration stage takes the recovered 3D structure and detected obstacles into a world co-ordinate system, and accumulates traversal cost over time using vehicle position and orientation information (from the VINS module described in Paragraph 5.2) to produce an obstacle map.

Team Autonomous Solutions has developed a LIDAR-based vehicle detector to augment the stereovision system, particularly at intersections where vehicles are viewed from arbitrary perspectives. Points in the LIDAR point cloud that are closely grouped and of the correct size for a vehicle are clustered and used to report the location of vehicles. A second LIDAR unit is used to eliminate ground clutter by rejecting low-slope surfaces that do not fit a vehicle profile.

Object information reported by the various stereo vision and LIDAR perception algorithms is fused into a single map. The Expert Driver assigns weights based on confidence and cost. For instance, remaining in lane lines is desired, but this rule can be violated to avoid collision with an obstacle or vehicle. If a collision appears to be inevitable, the robot should avoid the obstacles with highest confidence."

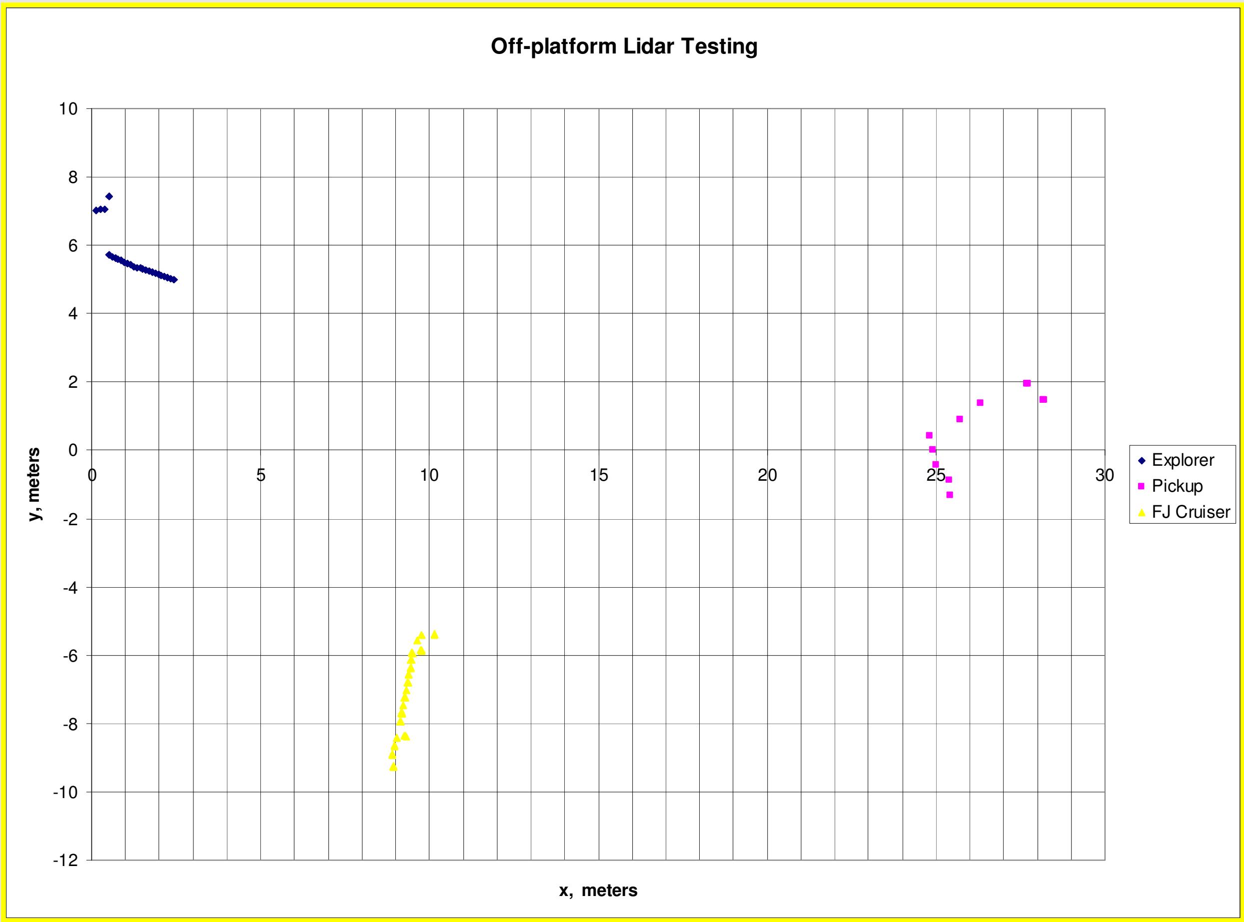

Figure 6, LIDAR Integration and Characterization Testing
The LIDAR sensor is located at the origin of the plot pointed along the positive x-axis. The points are vehicles detected by the LIDAR representing an intersection which demonstrates 180° by 30 m field of view.

5.1.2.2 Long-Range Vehicle Detection and Tracking

This module is responsible for detecting and tracking vehicles at long ranges as shown in the zones depicted in Figure 3, Sensor Placement. A stereo image pair is analyzed for vehicle like features using both image as well as depth cues. The features are then grouped together into bounding boxes that depict individual vehicles. Each detected object is passed through a series of false-positive (FP) removal gates to reject any spurious non-vehicle detections. As a final FP removal gate, an extensively trained AdaBoost based classifier is employed. Using this cascade of FP removal gates, we are able to achieve a very high detection rate while keeping the FP rate low. The system also tracks a detected vehicle across time using a combination of Kalman and Particle Filters to predict its motion and estimate its lateral and longitudinal speeds. The host vehicle motion, if available, readily plugs into the filters allowing even better tracking and velocity estimation. Figure 7, Long Range Vehicle and Lane Detection, shows a sample vehicle detection result from an experiment conducted on the Urban Challenge vehicle at Princeton, NJ.

Figure 7, Long Range Vehicle and Lane Detection

5.1.3 Stereo-based Lane detection

This module detects and tracks lanes up to 40m ahead of the vehicle. A set of high-contrast features is first selected from the input image. These features pass through a range based projection system that allows only features present on the ground to pass through. The features deemed to be present on the ground are then grouped together into individual lane-segments using a combination of contrast and geometric cues. The system uses host vehicle motion available from the CAN bus to further refine the lane-detection by integrating it across time.

5.2 VINS (Position Estimation)

The Sarnoff Video Inertial Navigation System (VINS) couple GPS, an inertial measurement unit, and multiple cameras to perform localization in both GPS-available and GPS-denied areas. Detailed descriptions of the VINS solution may be found in [2]. A brief description is provided in this section. Central to the Sarnoff VINS solution is our Multi-camera Visual Odometry algorithm. This algorithm estimates camera pose from image sequences and employs the following steps:

1. Acquire images from the left and right cameras at time t_k
2. Detect and match feature points in each stereo pair; use epipolar and disparity constraints to eliminate false matches
3. Compute 3D locations corresponding to these feature points using stereo triangulation
4. Perform 2D-2D image feature matching over time to establish 3D-2D point correspondences
5. Estimate camera pose using a robust resection method based on RANSAC followed by iterative refinement of the winning hypothesis

The Visual Odometry algorithm only provides relative pose estimates. For absolute location and orientation information, the Visual Odometry output is combined with GPS and IMU information as shown in Figure 8, VINS Block Diagram.

Figure 9, Vehicle path recovered by Sarnoff VINS during experiments conducted in Trenton, NJ and Figure 10, Vehicle path recovered by Sarnoff VINS during experiments conducted in Princeton, NJ show the VINS performance in urban environments that have GPS-denied areas. For the purposes of these experiments, GPS signals were manually suppressed to show the drift performance of the Multi-camera Visual Odometry. Experiments have been shown that even when GPS signals are completely denied, VINS localization drifts only 0.5% - 1% of the distance traveled.

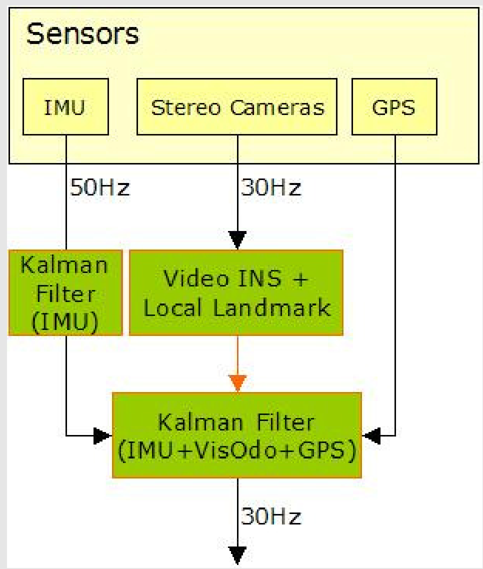

Figure 8, VINS Block Diagram

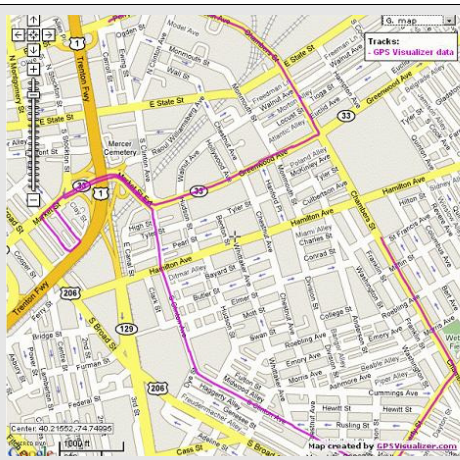

Figure 9, Vehicle path recovered by Sarnoff VINS during experiments conducted in Trenton, NJ

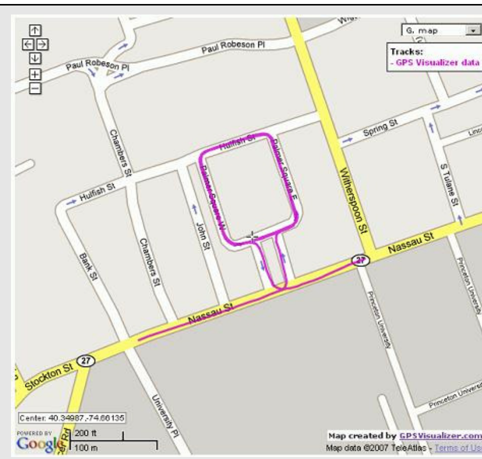

Figure 10, Vehicle path recovered by Sarnoff VINS during experiments conducted in Princeton, NJ

5.3 Mobius™ (High Level Planning)

Mobius™ is a software interface for controlling and monitoring multiple unmanned systems using the JAUS protocol. Mobius™ includes a mission planning component that performs high level vehicle route planning where paths produced are restricted by the constraints of the vehicle (minimum turning radius, width, etc.). Mobius™ processes the RNDF and generates a high level plan from the MDF using an adapted A* algorithm to reach all of the checkpoints for the autonomous vehicle. The plan will have path information which includes a minimum and maximum velocity for the path segments as specified in the MDF. This initial path will handle 3 point turns, U-turns, stops, parking, staying in predetermined lanes, and lane changes through the use of path segments and actions. Once planned, the path is sent to the Mission Spooler to be driven by the Neuron™ vehicle controller and monitored and/or modified by the Reactive Planner.

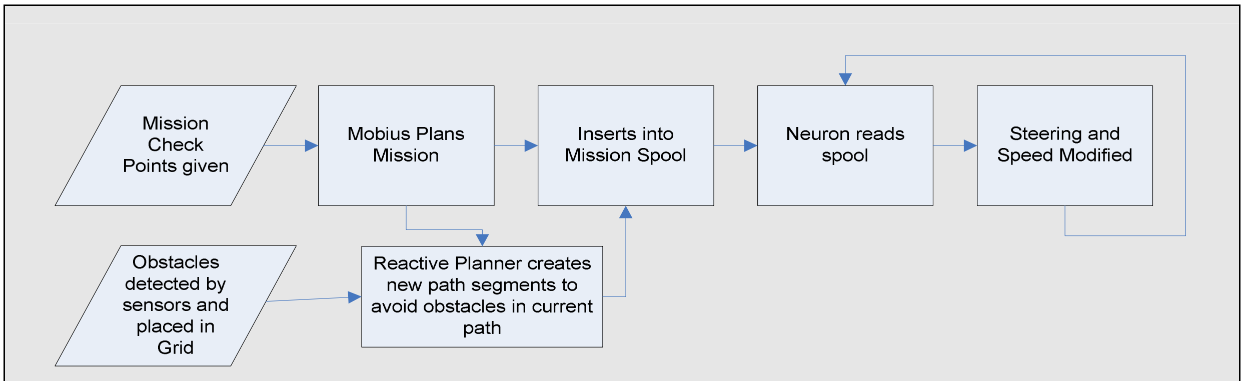

Figure 11, Mission Execution Sequence

Mobius™ is used as the primary simulation environment for software algorithm development and testing. Mobius™ can simultaneously command and monitor hundreds of vehicles, both real and\or simulated. This allows engineers to easily simulate complex traffic situations and analyze algorithm effectiveness. Mobius™ is also used as the primary testing and monitoring station during testing phases. In addition to Mobius™ running on board the vehicle, multiple test personnel can log in and monitor vehicle and mission performance via Mobius™ clients from remote locations over a wireless network. This feature greatly enhances our ability to test and collect data for analysis and allows us to remotely visualize and log test results from missions.

Figure 12, High Level Plan as generated by Mobius™

5.4 Reactive Planning

While the high level planner generates a route between checkpoints, the reactive planner avoids local obstacles and traffic. The reactive planner checks the high level plan, which is stored in the mission spooler, and makes any changes suggested by dynamic updates. Two subsystems provide this updated information to the Reactive Planner: the Perception and Expert Driver Packages.

The Reactive Planner functions by taking the perception information and overlaying the Global Planner's path onto it. Lane markers, vehicles, and general obstacles are reported by the Perception Package. The reactive planner uses this information to assign costs and place heading-oriented control points in a local map. If the path is clear, with no obstacles in the way, it will continue to drive the current path. If the current path is blocked or has high cost, the algorithm will search the grid for the lowest cost path which minimally adjusts the current path. An A* search finds the optimal sequence of control points to get from the vehicle's current location to the next place it can reattach to the high level route. The sequence of control points is converted into a smooth series of path segments, which replace the segments in the mission spooler. Lane information provided by the Perception Package is used to update the grid with low-cost "obstacles" to discourage the vehicle from changing lanes.

The Reactive Planner also receives information from the Expert Driver. This module provides situational awareness and a real-time interpretation of road rules and highway laws. Applicable situations detected by the Expert Driver change the behavior of the Reactive Planner's search to generate paths that conform to requirements of safety, highway laws, and the rules of the competition.

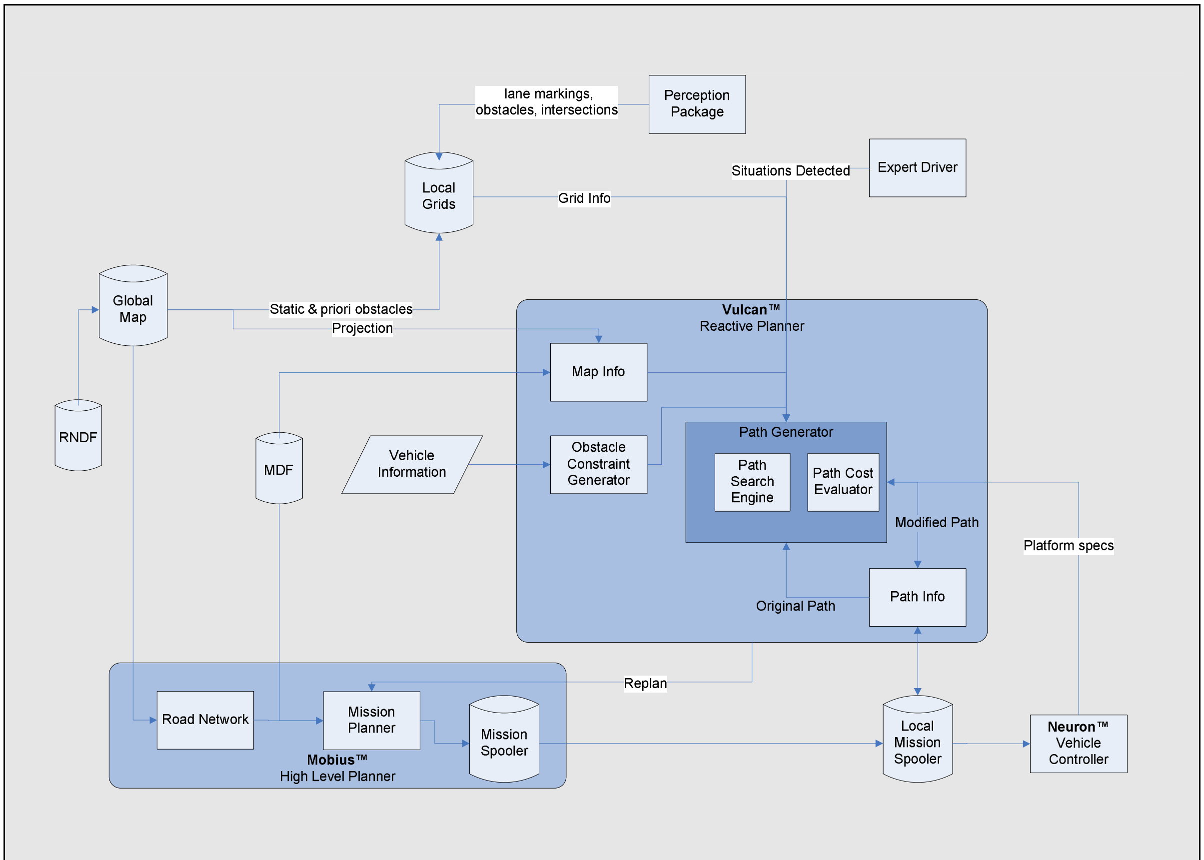

Figure 13, Reactive Planner Block Diagram

A completely blocked route is detected when the Reactive Planner search engine is unable to locally modify to the current path in a way that safely avoids the static obstacle and the anticipated route of the dynamic obstacles. Upon verification from the Expert Driver that a re-plan is necessary, a re-plan request is sent to the High Level Planner for re-routing. The High Level Planner will then create a new global path plan for the vehicle to follow. This same re-routing request would be called in the case where DARPA must move the vehicle and resume the mission from a different location.

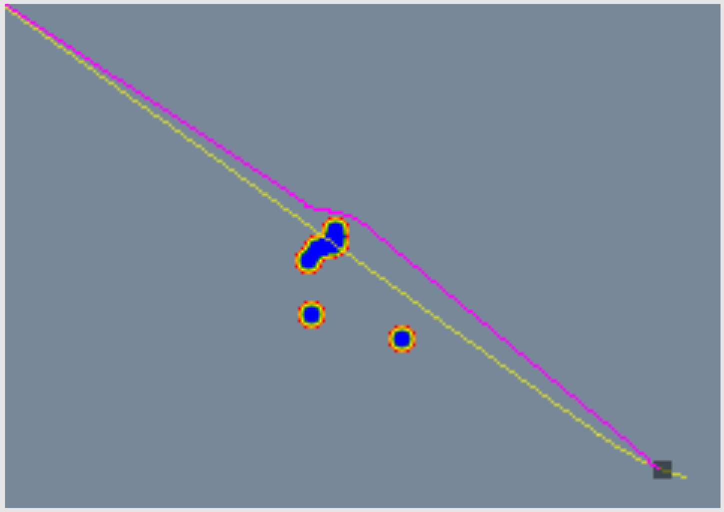

Figure 14, Reactive Planning Example

5.5 Expert Driver

While urban driving is a newer challenge for the robotics field, on-highway driving is a more developed regime. Past work in this area has developed models for generating situationally aware behaviors. Because decision trees and other monolithic models require every possible situation be enumerated and defined, these approaches are brittle and require large amounts of data. A better approach builds complex behaviors from a series of simple, independent rules, like the PolySAPIENT model [1].

Team Autonomous Solutions uses an expert driver that reduces complexity in the implementation through the introduction of independent reasoning objects and an arbiter function. The reasoning objects make recommendations to the arbiter based on their specific relationship between the autonomous vehicle and objects detected in the environment. These objects, or processing elements, are instantiated based on the current vehicle plan. The number of reasoning objects active at any given time is based on the complexity of the plan and the environment. The arbiter tallies the recommendations from the various reasoning objects and generates the appropriate mobility command to the Neuron™ processor.

Two reasoning objects used by the Team Autonomous Solutions expert driver prevent crossing the yellow line when there is oncoming traffic and determine whether a vehicle in front of the robot is stalled. Since these two reasoning objects make independent decisions, they represent four possible driving situations as represented in the following table.

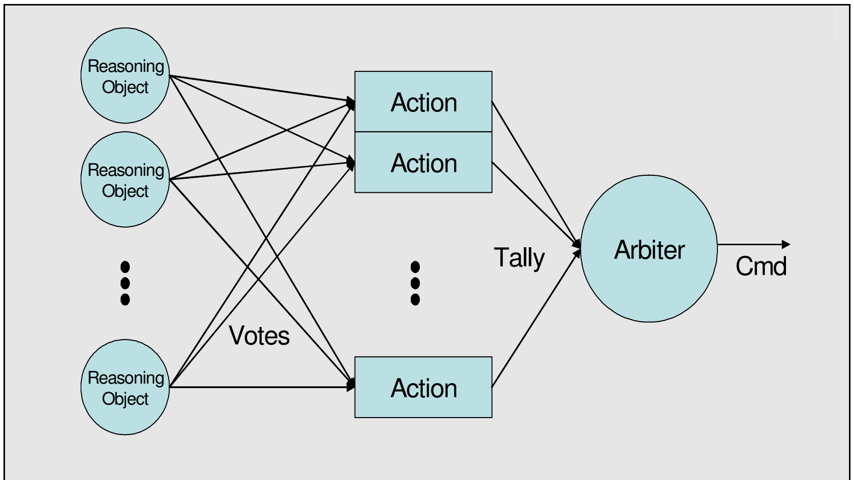

Figure 15, Expert Driver Block Diagram

Oncoming Traffic	Stalled Vehicle Ahead	Behavior from A	Behavior from B	Final Behavior
True	True	Don't cross yellow	Cross yellow	Don't cross
True	False	Don't cross yellow	Don't cross yellow	Don't cross
False	True	OK to cross	Cross yellow	Do cross
False	False	OK to cross	Don't cross yellow	Don't cross

Figure 16, Expert Driver Decision Processing Example

Due to the independent natures of these local reasoning objects, the complexity of the system behaviors, different detectable situations, increases as the square of the number of reasoning objects. Therefore, a large number of complex situations are represented by a much smaller number of hand-coded reasoning objects. The number of behavioral rules that must be coded under the independent reasoning regime is the square root of the rules required for a Bayesian Decision Network or other monolithic model. Additionally, this model is robust to new and unexpected situations.

5.6 Neuron™ (Vehicle Controller)

The ruggedized Neuron™ Vehicle Control Unit (VCU) is based on the automotive Motorola 565 processor due to its ability to provide the necessary processing power, memory, and I/O capacity needed for autonomous operation of unmanned vehicles. The Neuron™ software commands the low level hardware of the vehicle as well as interfaces to the Mobius™ software. It includes the control algorithms necessary to control the unmanned vehicle at speeds up to 30 mph on both paved and unpaved road surfaces.

5.6.1 Path Controller

The Low-Level Controller receives path segments from the Mission Spooler and issues low-level commands to the vehicle's actuators to maintain desired velocity and position on the planned path to arrive at the high level waypoints. Initially, the vehicle's behavior was simulated using

Matlab and Simulink to model the vehicle and to refine Neuron™ so that it functions well with the Toyota Highlander Hybrid. After successful simulation, the path controls were refined on the Highlander using Mobius™ to drive the test scenarios.

The standard kinematic model for an Ackerman vehicle is:

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = \frac{v}{L}\tan\phi$$

The two control inputs are the forward speed v and the steering wheel angle ϕ. (x, y) are the components of the vehicle position, and θ is the vehicle heading. L is the wheelbase of the vehicle.

Two errors can be associated with a vehicle attempting to follow a desired path. These are the heading error and the orthogonal error. The heading error is the angle between the vehicle and the direction of the path at the orthogonal point. The orthogonal error is the shortest distance from the center point on the driven axle of the vehicle to the desired path. These two errors will be used to determine the correction to the steering angle. A third error is the speed error, but this is not associated with the physical path; each path segment will include a speed specification, and it will be the speed controller's function to drive the speed error to zero.

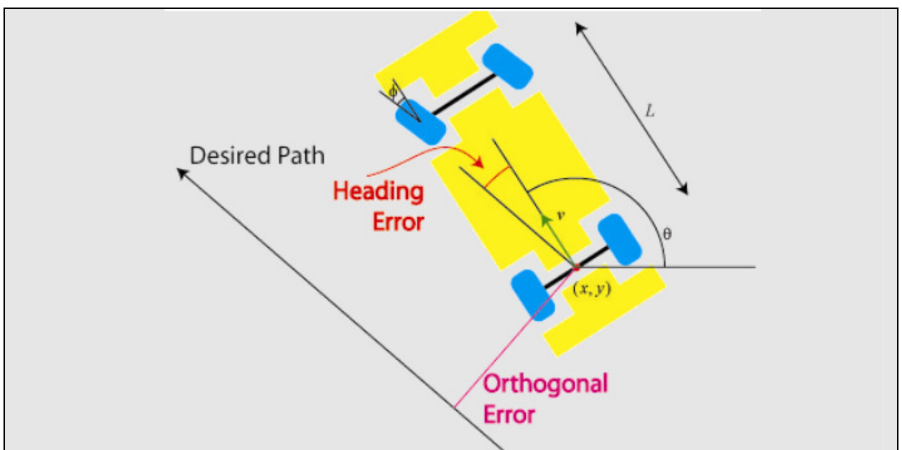

Figure 17, Errors Associated with Vehicle Control

The above describes a simple kinematic model. However, dynamic effects must also be included for effective control. This requires the use of additional vehicle parameters, such as the vehicle mass and inertia. Autonomous Solutions has in the past controlled lateral vehicle dynamics (steering control) using an approach developed at the University of California at Berkeley [3] with good success.

Figure 18, Simplified Vehicle Control System, shows a steering controller designed in Matlab/Simulink taking into account the kinematic and dynamic effects described above.

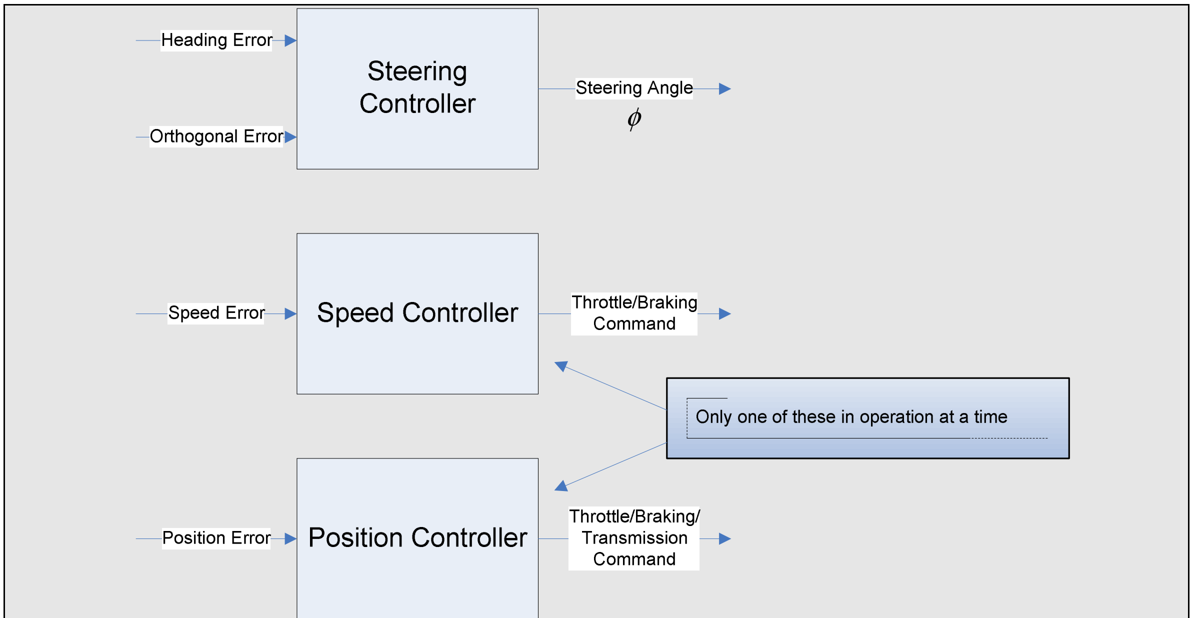

Figure 18, Simplified Vehicle Control System

5.6.2 Speed Controller

To account for the vehicle longitudinal (forward) dynamics, as well as a throttle actuator, experiments were performed to measure the vehicle speed in response to step commands in throttle position. This allowed the development of a speed controller and a position controller.

The speed controller is in operation in most cases but in certain circumstances will need to be replaced by a position controller. For example,

- The vehicle occasionally needs to stop at a precise location. This will be the case when it is parking.

- The vehicle may need to move forward in small intervals at an intersection to increase its view on either side

Figure 18, Simplified Vehicle Control System, depicts both speed and position controllers. Note that only one is in operation at a given time.

6 Results and Performance

Team Autonomous Solutions has established a suite of tests to exercise the capabilities as specified and described in the DARPA Technical Evaluation Criteria (June 12, 2006). This document lists the criteria under four categories, Basic Navigation, Basic Traffic, Advanced Navigation, and Advanced Traffic. Each of the specific capabilities within the categories is included in one or more tests. For the purposes of describing the current results and performance of the team's entry the test addressing the integration of functionality of leaving a lane to pass a parked vehicle is used. This test is designed to demonstrate and measure the capability to identify and pass a stalled vehicle. Variations on the test are used to ensure performance under the expected conditions. These conditions are:

1. The traffic vehicle is stalled,
2. The traffic vehicle moves within 10 seconds, and
3. The traffic vehicle does not stop.

In addition to the standard performance checks such as pre-run check out and load of RNDF and MDF, data such as the perception system's detection of the traffic vehicle, the reactive planner's command to slow the vehicle, the queuing behavior, and meeting the prescribed minimum vehicle separations are recorded. The overall mission plan is represented in Figure 19, , showing the Mobius™ interface and the prescribed route. The vehicle's position at the time it encountered the stalled vehicle is shown.

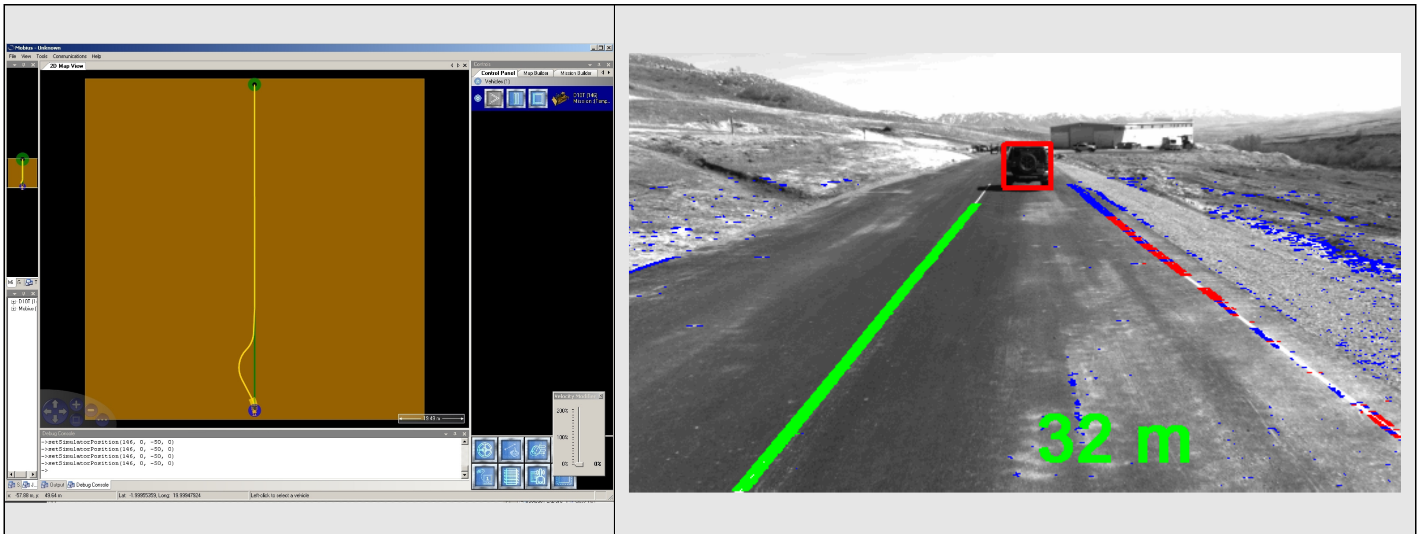

Figure 19, Vehicle Detection and Re-planning
Left is complete Mobius screen shot of Re-Planning and the right image Video overlay of Obstruction Prior to Re-Plan

To account for the detection of the stalled vehicle by the perception system, the test team captured screen shots of the output to the debug screens and recorded vehicle state data. The debug screen shots below show the overlay of the obstructed areas on the actual image captured from the driving camera and a two-dimensional view of the grid area directly in front of the vehicle.

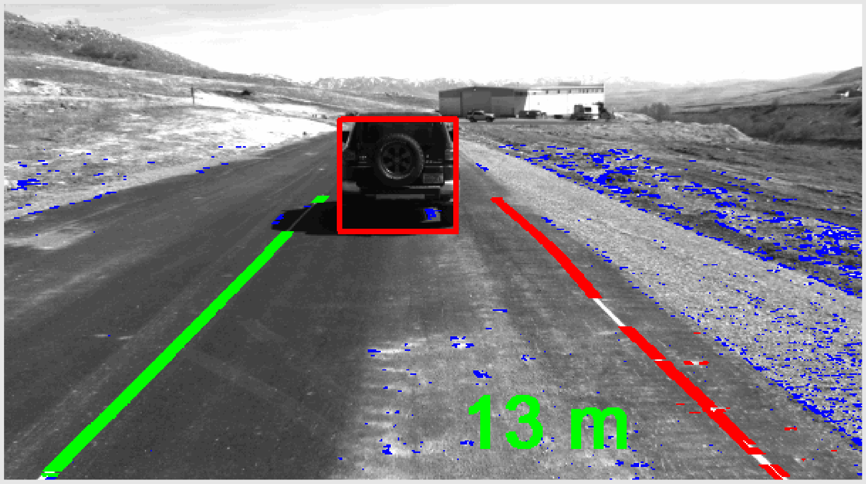

Figure 20, Video Overlay of Vehicle (Detected and Slowing)

When a vehicle is detected ahead in the same lane as the robot, the default behavior is to maintain safe gap maintenance. The reactive planner generates a geometric route that stays in the lane and passes through the lead vehicle. The expert driver then generates a braking curve that will slow the robot to match speed with the lead vehicle.

In this case, the lead vehicle is not moving, so the expert driver brings the robot to a complete stop. After a designated time has passed without motion, the expert driver determines that the lead vehicle is stalled and should be passed. The expert driver informs the reactive planner of this new state, and the reactive planner generates a plan that passes the stalled vehicle on the left as shown in Figure 21, Graphic of Reactive Plan Output.

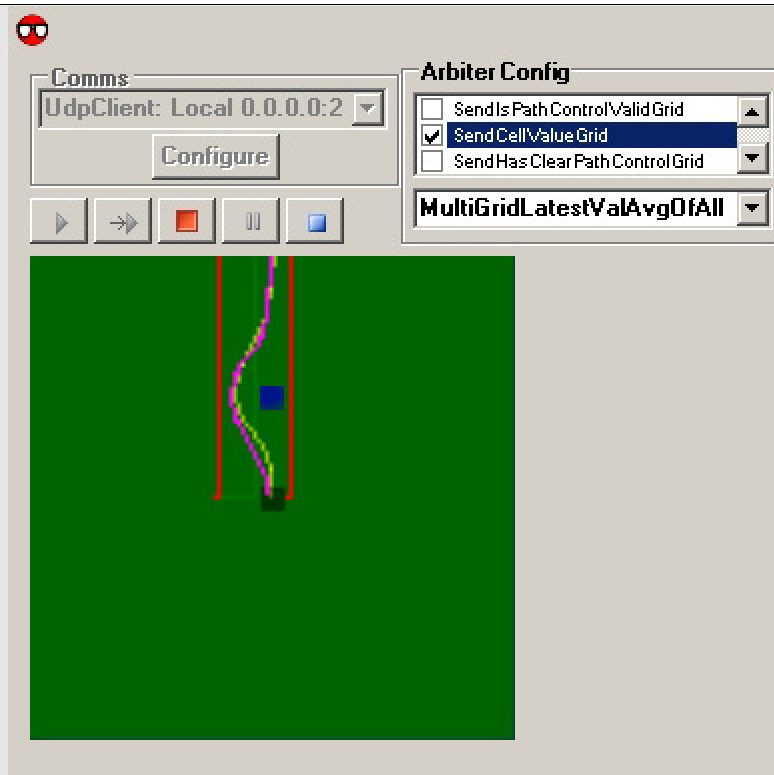

Figure 21, Graphic of Reactive Plan Output

The approach to testing by the team is to establish a few significant test cases that can be repeated on a weekly basis. This approach aids in the measurement of team performance against the schedule and is a good indicator of overall status. Variations on the core tests are used in support of integration of significant code changes. Regression testing, after integration, requires successful execution of all test cases.

7 Conclusion

Team Autonomous Solutions has conducted numerous tests in preparation for the Milestone 2 event tentatively scheduled for June 2007 near Salt Lake City Utah. To date successful demonstrations of lane line following, intersection precedence, and object detection and avoidance have been performed in the test vehicle systems. The majority of testing of complex behaviors and missions is performed in the simulation environment provided by Mobius™.

Test and evaluation of the autonomous system, as well as changes to the software as deemed necessary through testing, will continue up to the date of the Urban Challenge event. The current focus of the team is the Milestone 2 event. This event will include evaluation of the system for basic navigation and basic traffic behaviors. The specific behaviors are listed, with explanation, in the DARPA Technical Evaluation Criteria [6]. Team Autonomous Solutions' design as presented herein supports all requirements for the Milestone 2 Criteria.

8 References

1 Rahul Sukthankar, Situation Awareness for Tactical Driving, January 27, 1997, CMU-RI-TR-97-08, Robotics Institute Carnegie Mellon University, Pittsburgh, PA 15213,

2 David Nister, Oleg Naroditsky, James Bergen, Sarnoff Corporation, CN5300, "Visual Odometry for Ground Vehicle Applications", Princeton, New Jersey 08530, Journal of Field Robotics, January 2006, Wiley Volume 23, Issue 1

3 C. Taylor, J. Košecká, R. Blasi, J. Malik, "A Comparative Study of Vision-Based Lateral Control Strategies for Autonomous Highway Driving," *International Journal of Robotics Research,* 18:5, 1999.

4 A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen, "Real-time, multi-perspective perception for unmanned ground vehicles", in AUVSI 2003

5 A. Rieder, B. Southall, G. Salgian, R. Mandelbaum, H. Herman, P. Rander, T. Stentz, "Stereo perception on an off-road vehicle", in Intelligent Vehicle Symposium, 2002. IEEE

6 Defense Advanced Research Projects Agency, Urban Challenge Technical Evaluation Criteria, June 12, 2006,